

1  
2 **In the Claims**

3  
4 Claims 1, 9, 13, 21, 23, 29, 31 and 34 are amended.

5 Claims 1-38 remain in the application as are listed below:

6  
7 1. (Currently Amended) A system for synchronizing a visualization  
8 with audio samples comprising:

9 one or more audio sources configured to provide audio samples that are to  
10 be rendered by a media player;

11 an audio sample pre-processor communicatively linked with the one or  
12 more audio sources and configured to receive and pre-process audio samples  
13 before the samples are rendered, the pre-processing providing characterizing data  
14 associated with each sample, wherein the characterizing data is derived from the  
15 audio samples; and

16 one or more effects configured to receive the characterizing data and use  
17 the characterizing data to render a visualization that is synchronized with an audio  
18 sample that is being rendered by the media player.

19  
20 2. (Original) The system of claim 1 further comprising multiple data  
21 structures configured to hold the characterizing data, each data structure being  
22 associated with an audio sample.

23  
24 3. (Original) The system of claim 2, wherein the audio sample pre-  
25 processor is configured to maintain the data structures.

1  
2 4. (Original) The system of claim 2, wherein the audio sample pre-  
3 processor comprises a timestamp module that provides a timestamp for each audio  
4 sample, each timestamp being maintained by a data structure associated with the  
5 audio sample.

6  
7 5. (Original) The system of claim 4, wherein the timestamp is assigned  
8 by the timestamp module based upon when the audio sample is calculated to be  
9 rendered by the media player.

10  
11 6. (Original) The system of claim 4, wherein the audio sample pre-  
12 processor is configured to:

13 query a media player audio sample renderer for a time associated with an  
14 audio sample that is being currently rendered, and

15 use the time to ascertain a timestamp of an associated audio sample, the  
16 audio sample pre-processor further being configured to provide characterizing data  
17 of the associated audio sample so that the characterizing data can be used to render  
18 the visualization.

19  
20 7. (Original) The system of claim 1, wherein said characterizing data  
21 comprises frequency data.

22  
23 8. (Original) The system of claim 1, wherein said audio sample pre-  
24 processor comprises a Fast Fourier Transform that it utilizes to process the audio  
25 samples to provide frequency data associated with the audio samples.

1  
2 9. (Currently Amended) A media player comprising:

3 an audio sample pre-processor configured to receive and pre-process audio  
4 samples before the samples are rendered by the media player, the pre-processing  
5 providing frequency data associated with each sample, wherein the frequency data  
6 is derived from the audio samples; and

7 one or more effects configured to receive the frequency data and use the  
8 frequency data to render a visualization that is synchronized with an audio sample  
9 that is being rendered by the media player.

10  
11 10. (Original) The media player of claim 9 further comprising multiple  
12 data structures configured to hold the frequency data, each data structure being  
13 associated with an audio sample.

14  
15 11. (Original) The media player of claim 10, wherein the audio sample  
16 pre-processor comprises a timestamp module that provides a timestamp for each  
17 audio sample, each timestamp being maintained by a data structure associated with  
18 the audio sample, and further wherein the audio sample pre-processor is  
19 configured to:

20 query a media player audio sample renderer for a time associated with an  
21 audio sample that is being currently rendered, and

22 use the time to ascertain a timestamp of an associated audio sample, the  
23 audio sample pre-processor further being configured to provide frequency data of  
24 the associated audio sample to the one or more effects so that the frequency data  
25 can be used to render the visualization.

1  
2 12. (Original) The media player of claim 9, wherein the audio sample  
3 pre-processor pre-processes the audio samples by using a Fast Fourier Transform  
4 to provide the frequency data.

5  
6 13. (Currently Amended) A system for synchronizing a visualization  
7 with audio samples comprising:

8 an audio sample pre-processor configured to receive and preprocess audio  
9 samples before the samples are rendered by a renderer that comprises part of a  
10 media player, the audio sample preprocessor preprocessing the samples to provide  
11 characterizing data ~~associated with~~ derived from each sample, the characterizing  
12 data comprising a timestamp associated with each audio sample, the timestamp  
13 being assigned in accordance with when the audio sample is calculated to be  
14 rendered by the renderer;

B1  
15 multiple data structures configured to hold the characterizing data, each  
16 data structure being associated with an audio sample;

17 an audio rendering object configured to call the audio sample pre-processor  
18 to ascertain the characterizing data associated with an audio sample that is  
19 currently being rendered by the renderer;

20 the audio sample pre-processor being configured to ascertain said  
21 characterizing data by querying the renderer for a time associated with the  
22 currently-rendered audio sample, and then using said time to identify a data  
23 structure having a timestamp that is nearest in value to said time; and

24 one or more effects configured to receive characterizing data that is  
25 associated with the data structure having the timestamp that is nearest in value to

1 said time, and use the characterizing data to render a visualization that is  
2 synchronized with the audio sample that is being rendered by the renderer.

3  
4 14. (Original) The system of claim 13, wherein the characterizing data  
5 comprises frequency data.

6  
7 15. (Original) The system of claim 13, wherein the audio sample pre-  
8 processor comprises a Fast Fourier Transform that it utilizes to process the audio  
9 samples to provide frequency data associated with the audio samples.

10  
11 16. (Original) The system of claim 13, wherein the visualization is  
12 rendered in a rendering area in which other media types can be rendered.

13  
14 17. (Original) The system of claim 16, wherein the other media types  
15 comprise a video type.

16  
17 18. (Original) The system of claim 16, wherein the other media types  
18 comprise a skin type.

19  
20 19. (Original) The system of claim 16, wherein the other media types  
21 comprise a HTML type.

22  
23 20. (Previously Presented) The system of claim 16, wherein the other  
24 media types comprise an animation type.

21. (Currently Amended) A system for processing audio samples comprising:

a timestamp module for assigning timestamps to audio samples that are to be rendered by a media player renderer;

a spectrum analyzer for processing the audio samples to ~~provide~~ derive frequency data ~~associated with~~ from the audio samples;

multiple data structures each of which being associated with an audio sample, the data structures each containing timestamp data and frequency data for its associated audio sample; and

the system being configured to use the timestamp data to ascertain a data structure associated with an audio sample that is currently being rendered by the media player renderer and provide the frequency data associated with that audio sample so that the frequency data can be used to render a visualization associated with that audio sample.

22. (Original) The system of claim 21, wherein the spectrum analyzer comprises a Fast Fourier Transform that is utilized to provide the frequency data.

23. (Currently Amended) A method of providing a visualization comprising:

receiving multiple audio samples;

pre-processing the audio samples before they are rendered by a media player renderer, the pre-processing ~~providing~~ deriving characterizing data ~~for~~ from each sample;

1 determining when an audio sample is being rendered by the media player  
2 renderer; and

3 responsive to said determining, using the characterizing data that is  
4 associated with the audio sample that is being rendered to provide a visualization.

5  
6 24. (Original) The method of claim 23 further comprising maintaining  
7 characterizing data for each audio sample in a data structure associated with each  
8 audio sample.

9  
10 25. (Original) The method of claim 24, wherein the characterizing data  
11 comprises a timestamp associated with the audio sample, the timestamp being  
12 provided based upon when the audio sample is calculated to be rendered by the  
13 media player renderer.

14  
15 26. (Original) The method of claim 25, wherein said determining  
16 comprises:

17 ascertaining a time associated with a currently-rendered audio sample;  
18 selecting a data structure having a timestamp that is nearest the time; and  
19 providing characterizing data associated with the selected data structure to a  
20 component configured to provide the visualization.

21  
22 27. (Original) The method of claim 23, wherein the characterizing data  
23 comprises frequency data associated with each sample.

1        28. (Original) The method of claim 23, wherein said pre-processing  
2 comprises using a Fast Fourier Transform to provide frequency data associated  
3 with the samples.

4  
5        29. (Currently Amended) A method of providing a visualization  
6 comprising:

7            receiving multiple audio samples;

8            pre-processing the audio samples before they are rendered by a media  
9 player renderer, the pre-processing comprising at least (1) using a Fast Fourier  
10 Transform to ~~provide~~ derive frequency data ~~associated with~~ from the samples, and  
11 (2) associating a timestamp with each sample;

12            maintaining frequency data and a timestamp for each sample in a data  
13 structure;

14            determining when an audio sample is being rendered by a media player  
15 renderer by:

16            ascertaining a time associated with a currently-rendered sample; and

17            selecting a data structure having a timestamp that is nearest the time; and

18            providing frequency data associated with the selected data structure to a  
19 component configured to use the frequency data to render the visualization.

20  
21        30. (Original) One or more computer-readable media having computer-  
22 readable instructions thereon which, when executed by a computer, cause the  
23 computer to implement the method of claim 29.



1           31. (Currently Amended) A method of providing a visualization  
2 comprising:

3           calling an audio sample pre-processor for characterizing data that is has  
4 been derived from an associated ~~with an~~ audio sample that is currently being  
5 rendered by a media player renderer;

6           calling the media player renderer for a time associated with a currently-  
7 rendered audio sample;

8           using the time to select a data structure containing characterizing data  
9 associated with the currently-rendered audio sample; and

10          providing the characterizing data to a component for rendering a  
11 visualization.

12  
13          32. (Original) The method of claim 31, wherein the characterizing data  
14 comprises frequency data associated with the audio samples.

15  
16          33. (Original) The method of claim 31, wherein the characterizing data  
17 comprises frequency data associated with the audio samples and generated by pre-  
18 processing the audio samples using a Fast Fourier Transform.

19  
20          34. (Currently Amended) One or more computer-readable media having  
21 computer-readable instructions thereon which, when executed by a computer,  
22 cause the computer to:

23          pre-process audio samples using a Fast Fourier Transform to provide derive  
24 from the audio samples frequency data, the audio samples being pre-processed  
25 before they are rendered by a media player renderer;

1 query for frequency data that is associated with an audio sample that is  
2 currently being rendered by the media player renderer;  
3 query for a time associated with the currently-rendered audio sample;  
4 use the time to select a data structure containing frequency data associated  
5 with the currently-rendered audio sample; and  
6 provide the frequency data to a component that uses the frequency data for  
7 rendering a visualization.  
8

9 35. (Previously Presented) A method of providing a visualization  
10 comprising:

11 defining a frame rate at which visualization frames of a visualization are to  
12 be rendered, the visualization frames being rendered from characterizing data that  
13 is computed from audio samples and which is used to create the visualization;

14 setting a threshold associated with an amount of time that is to be spent  
15 rendering a visualization frame;

16 monitoring the time associated with rendering individual visualization  
17 frames;

18 determining whether a visualization frame rendering time exceeds the  
19 threshold; and

20 providing an effective frame rate for rendering visualization frames that is  
21 longer than the defined frame rate if the determined visualization frame rendering  
22 time exceeds the threshold.  
23  
24  
25

36. (Original) The method of claim 35, wherein said providing comprises increasing a call interval associated with calls that are made to a visualization-rendering component.

37. (Original) The method of claim 35 further comprising modifying the effective frame rate so that the visualization frames are rendered at the defined frame rate.

38. (Previously Presented) One or more computer-readable media having computer-readable instructions thereon which, when executed by a computer, cause the computer to:

31  
set a threshold associated with an amount of time that is to be spent rendering a visualization frame for a given frame rate, said visualization frame being associated with a visualization that is rendered using characterizing data computed from audio samples, which characterizing data is used to create the visualization;

monitor the time associated with rendering individual visualization frames;

determine whether a visualization frame rendering time exceeds the threshold; and

provide an effective frame rate for rendering the visualization that is longer than the defined frame rate if the determined visualization frame rendering time exceeds the threshold.